10/073,851

Refine Search

M.B.

10/28/05

Search Results -

Term	Documents
(23 AND 24).PGPB,USPT,USOC,EPAB,JPAB,DWPI,TDBD.	12
(L23 AND L24).PGPB,USPT,USOC,EPAB,JPAB,DWPI,TDBD.	12

Database:
US Pre-Grant Publication Full-Text Database
US Patents Full-Text Database
US OCR Full-Text Database
EPO Abstracts Database
JPO Abstracts Database
Derwent World Patents Index
IBM Technical Disclosure Bulletins

Search: L25

Refine Search

Recall Text Clear Interrupt

Search History

DATE: Tuesday, October 25, 2005 Printable Copy Create Case

Set Name side by side	**Query**	**Hit Count**	**Set Name** result set
	DB=PGPB,USPT,USOC,EPAB,JPAB,DWPI,TDBD; PLUR=YES; OP=ADJ		
L25	l23 and L24	12	L25
L24	virtual machine$1	10655	L24
L23	l21 and L22	13	L23
L22	java and native interface$1	584	L22
L21	platform independent and native and code$1 and shar$ memor$	162	L21
L20	safe language near1 code$1	1	L20
L19	shar$ memor$ and native code$1 and safe code$1	1	L19
L18	shar$ memor$ and native code$1 and safe code41	0	L18
L17	native code$1 and safe code$1 and shar$ memory	1	L17
L16	l10 and L9	0	L16
L15	l10 and L14	0	L15
L14	allocating near3 address range	29	L14
L13	allocating adj 11	5	L13

L12	native language and safe language	1	L12
DB=USPT; PLUR=YES; OP=ADJ			
L11	l9 and L10	0	L11
L10	native language$1	851	L10
L9	safe language$1	41	L9
L8	native language adj code$1	15	L8
L7	safe language adj code$1	0	L7
L6	native language and safe language	0	L6
L5	native language and safe language and java	0	L5
L4	native language and safe language ad java	0	L4
L3	l1 and L2	930	L3
L2	shar$ adj memory	9573	L2
L1	address range	6767	L1

END OF SEARCH HISTORY

Subscribe (Full Service) Register (Limited Service, Free) Login

Search: ● The ACM Digital Library ○ The Guide

native code and platform independent and shared memory

THE ACM DIGITAL LIBRARY

Feedback Report a problem Satisfaction survey

Terms used
native code and **platform independent** and **shared memory**

Found **53,492** of **164,603**

Sort results by: relevance
Display results: expanded form

Save results to a Binder
Search Tips
□ Open results in a new window

Try an Advanced Search
Try this search in The ACM Guide

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next
Best 200 shown

Relevance scale

1 CRL: high-performance all-software distributed shared memory
K. L. Johnson, M. F. Kaashoek, D. A. Wallach
December 1995 **ACM SIGOPS Operating Systems Review , Proceedings of the fifteenth ACM symposium on Operating systems principles**, Volume 29 Issue 5
Publisher: ACM Press , ACM Press
Full text available: pdf(2.02 MB) Additional Information: full citation, references, citings, index terms

2 Fast detection of communication patterns in distributed executions
Thomas Kunz, Michiel F. H. Seuren
November 1997 **Proceedings of the 1997 conference of the Centre for Advanced Studies on Collaborative research**
Publisher: IBM Press
Full text available: pdf(4.21 MB) Additional Information: full citation, abstract, references, index terms

Understanding distributed applications is a tedious and difficult task. Visualizations based on process-time diagrams are often used to obtain a better understanding of the execution of the application. The visualization tool we use is Poet, an event tracer developed at the University of Waterloo. However, these diagrams are often very complex and do not provide the user with the desired overview of the application. In our experience, such tools display repeated occurrences of non-trivial commun ...

3 Process migration
September 2000 **ACM Computing Surveys (CSUR)**, Volume 32 Issue 3
Publisher: ACM Press
Full text available: pdf(1.24 MB) Additional Information: full citation, abstract, references, citings, index terms, review

Process migration is the act of transferring a process between two machines. It enables dynamic load distribution, fault resilience, eased system administration, and data access locality. Despite these goals and ongoing research efforts, migration has not achieved widespread use. With the increasing deployment of distributed systems in general, and distributed operating systems in particular, process migration is again receiving more attention in both research and product development. As hi ...

Keywords: distributed operating systems, distributed systems, load distribution, process migration

4 Portable resource control in Java
Walter Binder, Jane G. Hulaas, Alex Villazón
October 2001 **ACM SIGPLAN Notices , Proceedings of the 16th ACM SIGPLAN conference on Object oriented programming, systems, languages, and applications**, Volume 36 Issue 11
Publisher: ACM Press , ACM Press
Full text available: pdf(307.08 KB) Additional Information: full citation, abstract, references, citings, index terms

Preventing abusive resource consumption is indispensable for all kinds of systems that execute untrusted mobile coee, such as mobile object sytems, extensible web servers, and web browsers. To implement the required defense mechanisms, some support for resource control must be available: accounting and limiting the usage of physical resources like CPU and memory, and of logical resources like threads. Java is the predominant implementation language for the kind of systems envisaged here, even th ...

Keywords: Java, bytecode rewriting, micro-kernels, mobile object systems, resource control, security

5 Memory access and virtualization techniques for performance: Enabling unrestricted automated synthesis of portable hardware accelerators for virtual machines
Miljan VuletiĆ, Christophe Dubach, Laura Pozzi, Paolo Ienne
September 2005 **Proceedings of the 3rd IEEE/ACM/IFIP international conference on Hardware/software codesign and system synthesis CODES+ISSS '05**
Publisher: ACM Press
Full text available: pdf(167.52 KB) Additional Information: full citation, abstract, references, index terms

The performance of virtual machines (e.g., *Java Virtual Machines*---JVMs) can be significantly improved when critical code sections (e.g., *Java bytecode* methods) are migrated from software to reconfigurable hardware. In contrast to the compile-once-run-anywhere concept of virtual machines, reconfigurable applications lack portability and transparent SW/HW interfacing: applicability of accelerated hardware solutions is often limited to a single platform. In this work, we apply a virt ...

Keywords: accelerator, synthesis, virtual machine

6 Multitasking without comprimise: a virtual machine evolution
Grzegorz Czajkowski, Laurent Daynés
October 2001 **ACM SIGPLAN Notices , Proceedings of the 16th ACM SIGPLAN conference on Object oriented programming, systems, languages, and applications**, Volume 36 Issue 11
Publisher: ACM Press , ACM Press
Full text available: pdf(220.97 KB) Additional Information: full citation, abstract, references, citings, index terms

The multitasking virtual machine (called from now on simply MVM) is a modification of the Java virtual machine. It enables safe, secure, and scalable multitasking. Safety is achieved by strict isolation of application from one another. Resource control augment security by preventing some denial-of-service attacks. Improved scalability results from an aggressive application of the main design principle of MVM: share as much of the runtime as possible among applications and replicate everything el ...

Keywords: Java virtual machine, application isolation, native code execution, resource control

7 Techniques for obtaining high performance in Java programs
Iffat H. Kazi, Howard H. Chen, Berdenia Stanley, David J. Lilja
September 2000 **ACM Computing Surveys (CSUR)**, Volume 32 Issue 3
Publisher: ACM Press
Full text available: pdf(816.13 KB) Additional Information: full citation, abstract, references, citings, index terms

This survey describes research directions in techniques to improve the performance of programs written in the Java programming language. The standard technique for Java execution is interpretation, which provides for extensive portability of programs. A Java interpreter dynamically executes Java bytecodes, which comprise the instruction set of the Java Virtual Machine (JVM). Execution time performance of Java programs can be improved through compilation, possibly at the expense of portabili ...

Keywords: Java, Java virtual machine, bytecode-to-source translators, direct compilers, dynamic compilation, interpreters, just-in-time compilers

8 ARMI: an adaptive, platform independent communication library
Steven Saunders, Lawrence Rauchwerger
June 2003 **ACM SIGPLAN Notices , Proceedings of the ninth ACM SIGPLAN symposium on Principles and practice of parallel programming**, Volume 38 Issue 10
Publisher: ACM Press , ACM Press
Full text available: pdf(242.64 KB) Additional Information: full citation, abstract, references, index terms

ARMI is a communication library that provides a framework for expressing fine-grain parallelism and mapping it to a particular machine using shared-memory and message passing library calls. The library is an advanced implementation of the RMI protocol and handles low-level details such as scheduling incoming communication and aggregating outgoing communication to coarsen parallelism when necessary. These details can be tuned for different platforms to allow user codes to achieve the highest perf ...

Keywords: MPI, OpenMP, Pthreads, RMI, RPC, communication library, parallel programming, run-time system

9 Performance of hybrid message-passing and shared-memory parallelism for discrete element modeling
D. S. Henty
November 2000 **Proceedings of the 2000 ACM/IEEE conference on Supercomputing (CDROM)**
Publisher: IEEE Computer Society
Full text available: pdf(197.99 KB) Publisher Site Additional Information: full citation, abstract, references, citings, index terms

The current trend in HPC hardware is towards clusters of shared-memory (SMP) compute nodes. For applications developers the major question is how best to program these SMP clusters. To address this we study an algorithm from Discrete Element Modeling, parallelized using both the message-passing and shared-memory models simultaneously ("hybrid" parallelization). The natural load-balancing methods are different in the two parallel models, the shared-memory method being in princip ...

10 A framework for efficient reuse of binary code in Java
Pramod G. Joisha, Samuel P. Midkiff, Mauricio J. Serrano, Manish Gupta
June 2001 **Proceedings of the 15th international conference on Supercomputing**
Publisher: ACM Press
Full text available: pdf(419.49 KB) Additional Information: full citation, abstract, references, index terms

This paper presents a compilation framework that enables efficient sharing of executable code across distinct Java Virtual Machine (JVM) instances. High-performance JVMs rely on run-time compilation, since static compilation cannot handle many dynamic features of Java. These JVMs suffer from large memory footprints and high startup costs, which are serious problems for embedded devices (such as hand held personal digital assistants and cellular phones) and scalable servers. A recently propose ...

11 Application-level checkpointing for shared memory programs
Greg Bronevetsky, Daniel Marques, Keshav Pingali, Peter Szwed, Martin Schulz
October 2004 **Proceedings of the 11th international conference on Architectural support for programming languages and operating systems**, Volume 32 , 38 , 39 Issue 5 , 5 , 11
Publisher: ACM Press , ACM Press , ACM Press , ACM Press
Full text available: pdf(235.77 KB) Additional Information: full citation, abstract, references, index terms

Trends in high-performance computing are making it necessary for long-running applications to tolerate hardware faults. The most commonly used approach is checkpoint and restart (CPR) - the state of the computation is saved periodically on disk, and when a failure occurs, the computation is restarted from the last saved state. At present, it is the responsibility of the programmer to instrument applications for CPR.Our group is investigating the use of compiler technology to instrument codes to ...

Keywords: checkpointing, fault-tolerance, openMP, shared-memory programs

12 Supercomputers: Evaluating support for global address space languages on the Cray X1
Christian Bell, Wei-Yu Chen, Dan Bonachea, Katherine Yelick
June 2004 **Proceedings of the 18th annual international conference on Supercomputing**
Publisher: ACM Press
Full text available: pdf(265.56 KB) Additional Information: full citation, abstract, references, index terms

The Cray X1 was recently introduced as the first in a new line of parallel systems to combine high-bandwidth vector processing with an MPP system architecture. Alongside capabilities such as automatic fine-grained data parallelism through the use of vector instructions, the X1 offers hardware support for a transparent global-address space (GAS), which makes it an interesting target for GAS languages. In this paper, we describe our experience with developing a portable, open-source and high perfo ...

Keywords: UPC, X1, global address space

13 Multigrain shared memory
Donald Yeung, John Kubiatowicz, Anant Agarwal
May 2000 **ACM Transactions on Computer Systems (TOCS)**, Volume 18 Issue 2
Publisher: ACM Press
Full text available: pdf(369.18 KB) Additional Information: full citation, abstract, references, index terms, review

Parallel workstations, each comprising tens of processors based on shared memory, promise cost-effective scalable multiprocessing. This article explores the coupling of such small- to medium-scale shared-memory multiprocessors through software over a local area network to synthesize larger shared-memory systems. We call these systems Distributed Shared-memory MultiProcessors (DSMPs). This article introduces the design of a shared-memory system that uses multiple granularities of sharing, ca ...

Keywords: distributed memory, symmetric multiprocessors, system of systems

14 Sharing and protection in a single-address-space operating system
Jeffrey S. Chase, Henry M. Levy, Michael J. Feeley, Edward D. Lazowska
November 1994 **ACM Transactions on Computer Systems (TOCS)**, Volume 12 Issue 4
Publisher: ACM Press
Full text available: pdf(2.87 MB) Additional Information: full citation, abstract, references, citings, index terms

This article explores memory sharing and protection support in Opal, a single-address-space operating system designed for wide-address (64-bit) architectures. Opal threads execute within protection domains in a single shared virtual address space. Sharing is simplified, because addresses are context independent. There is no loss of protection, because addressability and access are independent; the right to access a segment is determined by the protection domain in which a thread executes. T ...

Keywords: 64-bit architectures, capability-based systems, microkernel operating systems, object-oriented database systems, persistent storage, protection, single-address-space operating systems, wide-address architectures

15 Using generative design patterns to generate parallel code for a distributed memory environment
Kai Tan, Duane Szafron, Jonathan Schaeffer, John Anvik, Steve MacDonald
June 2003 **ACM SIGPLAN Notices , Proceedings of the ninth ACM SIGPLAN symposium on Principles and practice of parallel programming**, Volume 38 Issue 10
Publisher: ACM Press , ACM Press
Full text available: pdf(385.41 KB) Additional Information: full citation, abstract, references, index terms

A design pattern is a mechanism for encapsulating the knowledge of experienced designers into a re-usable artifact. Parallel design patterns reflect commonly occurring parallel communication and synchronization structures. Our tools, CO2P3S (Correct Object-Oriented Pattern-based Parallel Programming System) and MetaCO2P3S, use *generative design patterns*. A programmer selects the parallel design patterns that are appropriate for an application, and then adapts the patterns for that specifi ...

Keywords: design patterns, frameworks, parallel programming, programming tools

16 Program transformation and runtime support for threaded MPI execution on shared-memory machines
Hong Tang, Kai Shen, Tao Yang
July 2000 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 22 Issue 4
Publisher: ACM Press
Full text available: pdf(352.21 KB) Additional Information: full citation, abstract, references, citings, index terms

Parallel programs written in MPI have been widely used for developing high-performance applications on various platforms. Because of a restriction of the MPI computation model, conventional MPI implementations on shared-memory machines map each MPI node to an OS process, which can suffer serious performance degradation in the presence of multiprogramming. This paper studies compile-time and runtime techniques for enhancing performance portability of MPI code running on multiprogrammed share ...

Keywords: MPI, lock-free synchronization, multiprogrammed environments, program transformation, shared-memory machines, threaded execution

17 Source-level global optimizations for fine-grain distributed shared memory systems
R. Veldema, R. F. H. Hofman, R. A. F. Bhoedjang, C. J. H. Jacobs, H. E. Bal
June 2001 **ACM SIGPLAN Notices , Proceedings of the eighth ACM SIGPLAN symposium on Principles and practices of parallel programming**, Volume 36 Issue 7
Publisher: ACM Press , ACM Press
Full text available: pdf(112.60 KB) Additional Information: full citation, abstract, references, citings, index terms

This paper describes and evaluates the use of aggressive static analysis in Jackal, a fine-grain Distributed Shared Memory (DSM) system for Java. Jackal uses an optimizing, source-level compiler rather than the binary rewriting techniques employed by most other fine-grain DSM systems. Source-level analysis makes existing access-check optimizations (e.g., access-check batching) more effective and enables two novel fine-grain DSM optimizations: object-graph aggregatio ...

18 Towards transparent and efficient software distributed shared memory
Daniel J. Scales, Kourosh Gharachorloo
October 1997 **ACM SIGOPS Operating Systems Review , Proceedings of the sixteenth ACM symposium on Operating systems principles**, Volume 31 Issue 5
Publisher: ACM Press , ACM Press
Full text available: pdf(2.34 MB) Additional Information: full citation, references, citings, index terms

19 The effectiveness of affinity-based scheduling in multiprocessor network protocol processing (extended version)
James D. Salehi, James F. Kurose, Don Towsley
August 1996 **IEEE/ACM Transactions on Networking (TON)**, Volume 4 Issue 4
Publisher: IEEE Press
Full text available: pdf(1.71 MB) Additional Information: full citation, references, citings, index terms

20 Object and native code thread mobility among heterogeneous computers (includes sources)
B. Steensgaard, E. Jul
December 1995 **ACM SIGOPS Operating Systems Review , Proceedings of the fifteenth ACM symposium on Operating systems principles**, Volume 29 Issue 5
Publisher: ACM Press , ACM Press
Full text available: pdf(1.50 MB) Additional Information: full citation, references, citings, index terms

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next